# Polymorphismus

Gehen wir von der Situation aus,

dass Sie mehrere Methoden haben,

die eine gleiche oder ähnliche Aufgabe erfüllen,

sich jedoch in der Anzahl der Parameter unterscheiden.

In unserem Beispiel wollen wir von 2, 3, 4 oder mehr übergebenen int-Parametern entscheiden,

ob sie aufsteigend sortiert sind.

Dazu implementieren wir Methoden,

in deren Namen wir die Anzahl der Parameter einfließen lassen:

"sort\_2()", "sort\_3()" …

Gedanke 1:

Was stört uns dabei:

1. Die Vorgehensweise birgt die Gefahr,

dass wir uns vertun,

z.B. die Methode "sort\_2()" versehentlich mit 3 Parametern aufrufen

1. Wir haben Aufwand,   
   die Methoden unterschiedlich zu benennen.

Gedanke 2:

Wir überlegen uns,

wie der Compiler beim Aufrufen von Methoden vorgeht:

Findet er im Hauptprogramm einen Methodenaufruf,

zu erkennen an dem runden Klammerpaar,

sucht er nach der Hauptmethode,

wo die Deklaration der Methode steht.

Dabei vergleicht er den Methodennamen, den er sucht,

nacheinander mit den Methodendeklarationen.

Aus diesen beiden Gedankengängen heraus fragen wir uns,

ob wir uns nicht die Arbeit vereinfachen könnten

u. diese auf den Compiler abwälzen in der Form,

dass wir alle Methoden gleich nennen

u. dem Compiler die Zuweisung überlassen.

Er soll, wenn er nach der passenden Methodendeklaration sucht,

nicht nur schauen, dass der Methodenname stimmt,

sondern dass die Parameterlisten übereinstimmen.

Dass das so funktioniert, ist die Aussage des Polymorphismus.

Das Wort bedeutet wörtlich übersetzt "Vielgestaltigkeit":

Eine Methode mit einem Namen

tritt in vielen Gestalten auf

in Form von Ausprägungen mit unterschiedlichen Paramterlisten.

Aussage des Polymorphismus:

Es darf mehrere Methoden gleichen Namens parallel geben,

sofern sie sich in der Parameterliste unterscheiden.

Der Compiler sucht bei Aufruf einer Methode

anhand der aktuellen Parameterliste

die Deklaration der Methode heraus,

die zur formalen Parameterliste passt.

Wie dürfen sich die Parameterlisten unterscheiden?

1. Länge: Z.B. eine Parameterliste hat Länge 0, die andere 1.
2. Typ: Bei gleicher Länge sind unterschiedliche Typen zulässig,  
   z.B. bei jeweils der Länge 1 ein int u. ein bool.
3. Reihenfolge: Z.B. erst ein int u. dann ein bool  
   u. als zweites erst ein bool u. dann ein int.

Eine unterschiedliche Benennung der Paramter

bei gleicher Länge, gleichem Typ u. gleicher Reihenfolge

ist kein Unterscheidungsmerkmal:

Es erfolgt die Fehlermeldung: "Doublicate method declaration…"